

Fourth Communications

Lab2_organization

Sec (1):

1- Asmaa Mohamed Kamal

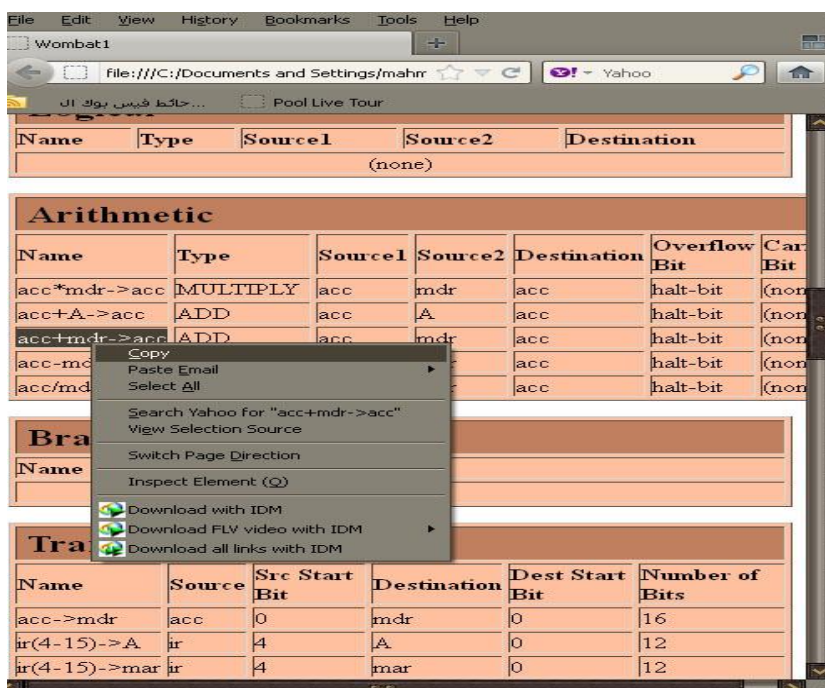
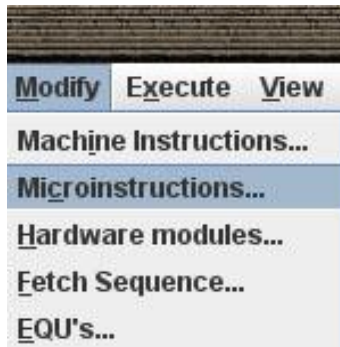
2- Asmaa Abdelmonem Eyssa

3- Asmaa Shaban Mahmoud

4- Basma Gamal

ANSWER QUESTION (1):

Add ACC to the MDR



Edit Microinstructions

Type of Microinstruction: **Arithmetic**

name	type	source1	source2	destination	overflowBit	carryBit
acc+mdr...	ADD	acc	mdr	acc	halt-bit	(none)

New Delete Duplicate

Help OK Cancel

test ACC if not zero , Skip the next microinstruction

Modify **Execute** **View**

Machine Instructions...

Microinstructions...

Hardware modules...

Fetch Sequence...

EQU's...

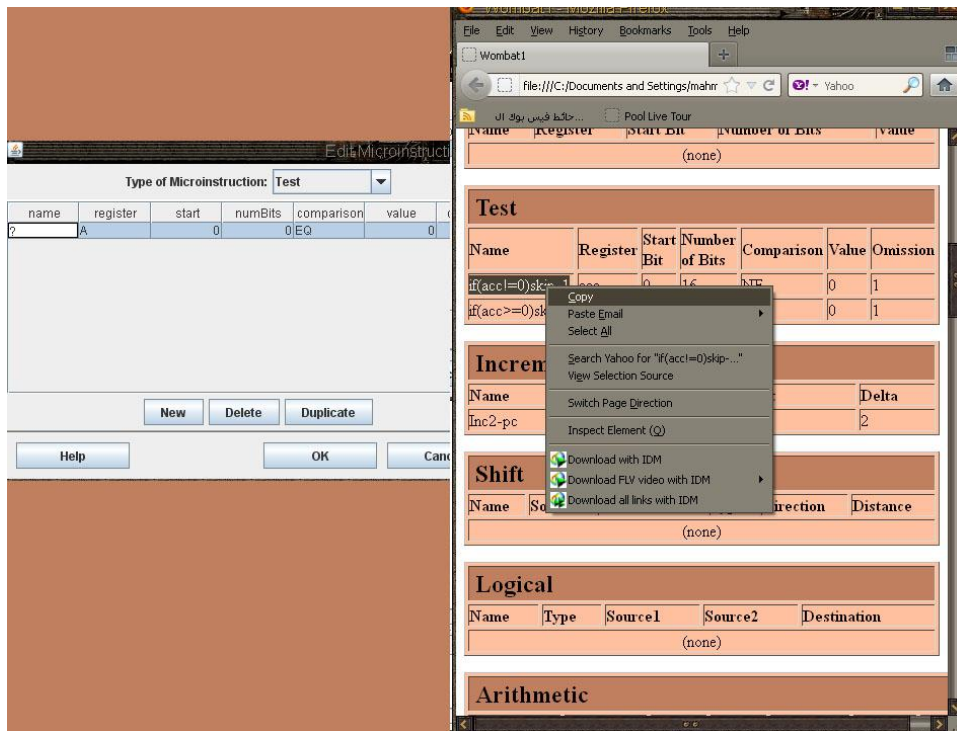
Edit Microinstructions

Type of Microinstruction: **Test**

name	register	start	numBit	value	omission
------	----------	-------	--------	-------	----------

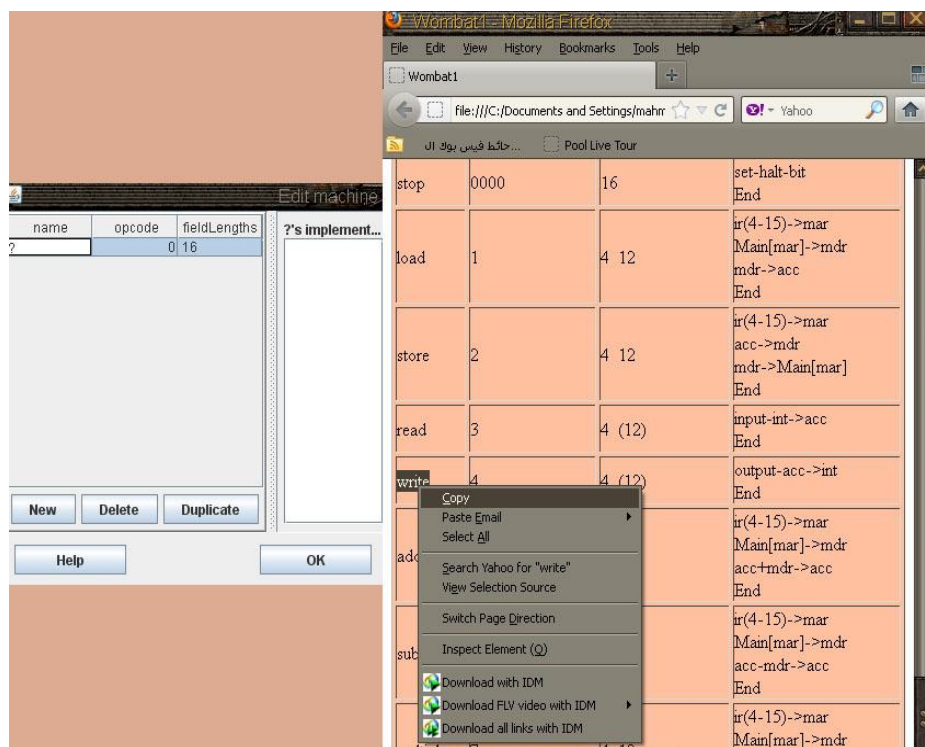
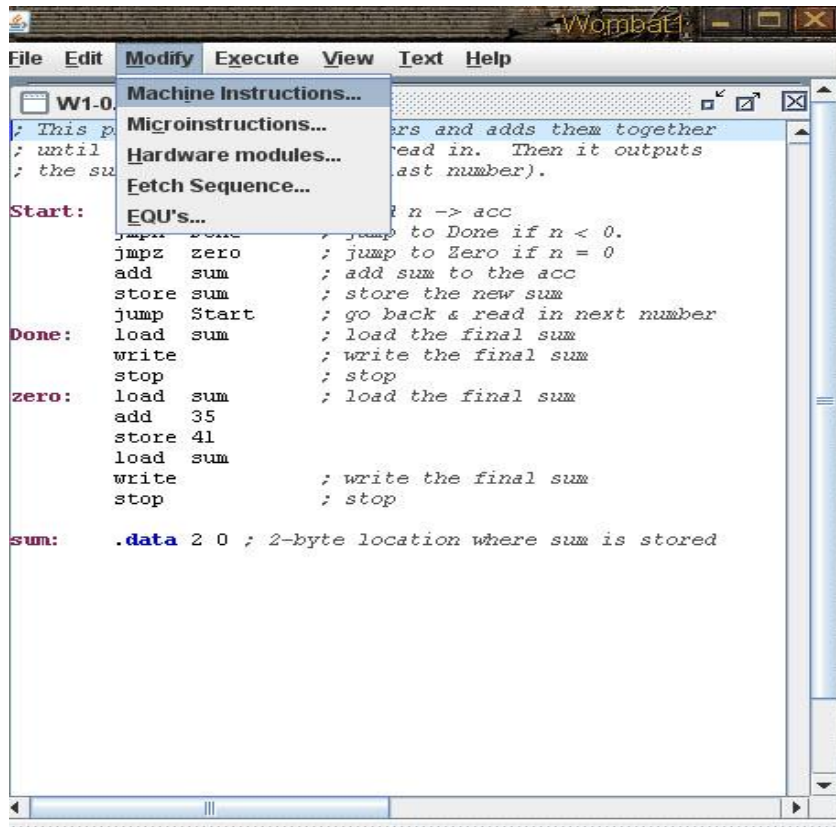
New Delete Duplicate

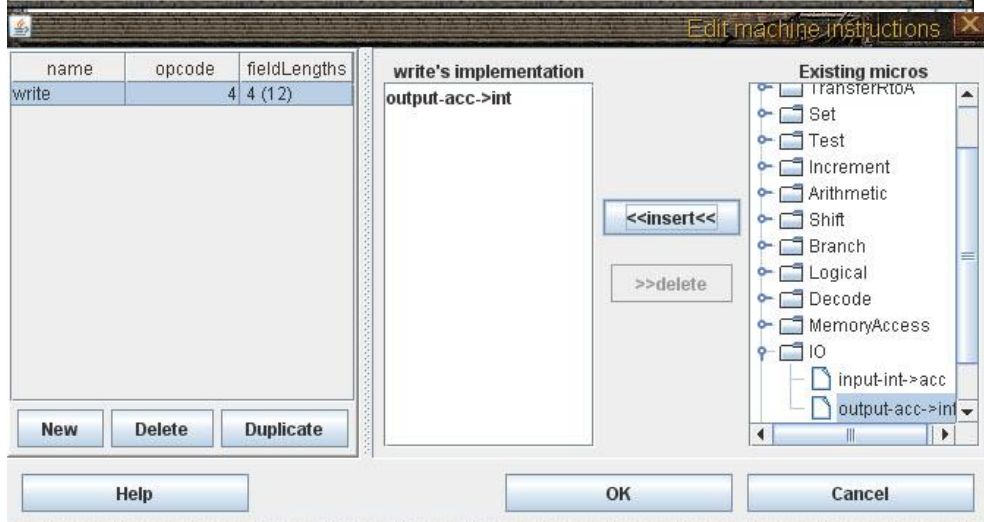
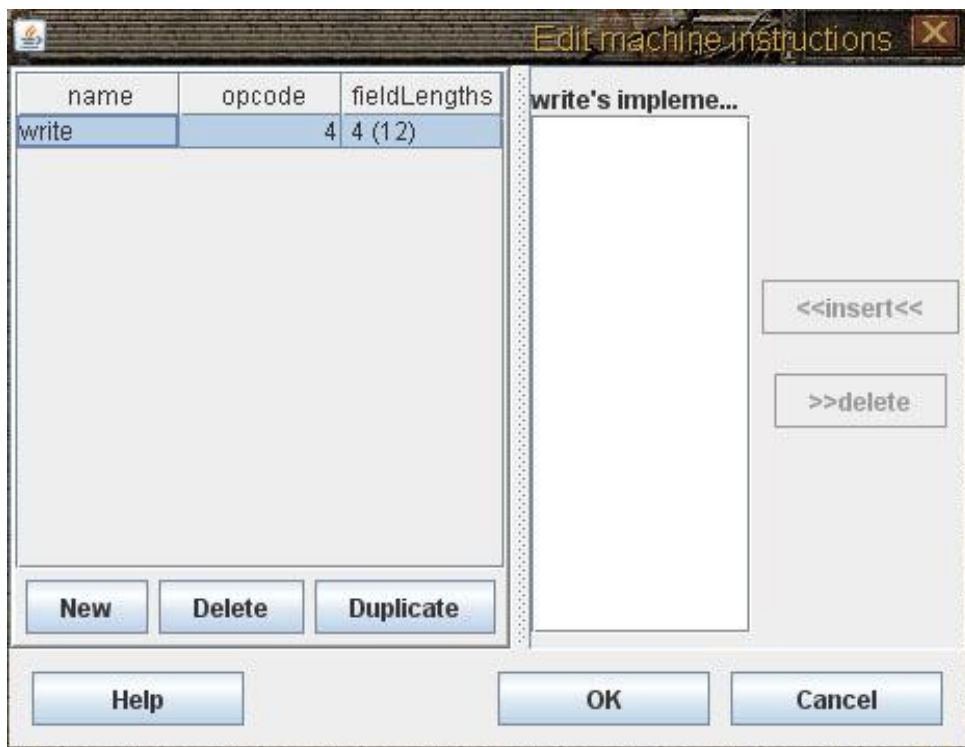
Help OK Cancel



ANSWER QUESTION (2):

Microinstruction Write





SIMILARLY

Microinstruction Divide

The screenshot shows the 'Edit machine instructions' dialog box. On the left, a table lists microinstructions with their names, opcodes, and field lengths. The 'divide' instruction is highlighted. In the center, the 'divide's implementation' is shown as a sequence of microinstructions: 'ir(4-15)->mar', 'Main[mar]->mdr', 'acc/mdr->acc', and 'End'. On the right, a list of 'Existing micros' is shown, including 'TransferRtoR', 'TransferAtoR', 'TransferRtoA', 'Set', 'Test', 'Increment', 'Arithmetic', 'Shift', 'Branch', 'Logical', 'Decode', 'MemoryAccess', 'IO', and 'SetCondBit'. Buttons for '<<insert<<', '>>delete', 'New', 'Delete', 'Duplicate', 'Help', 'OK', and 'Cancel' are visible.

name	opcode	fieldLengths
Add	C	4 12
stop	0	16
load	1	4 12
store	2	4 12
read	3	4 (12)
write	4	4 (12)
add	5	4 12
subtract	6	4 12
divide	8	4 12
jump	9	4 12
jmpz	A	4 12
jmpn	B	4 12

divide's implementation

```
ir(4-15)->mar
Main[mar]->mdr
acc/mdr->acc
End
```

Existing micros

- TransferRtoR
- TransferAtoR
- TransferRtoA
- Set
- Test
- Increment
- Arithmetic
- Shift
- Branch
- Logical
- Decode
- MemoryAccess
- IO
- SetCondBit

Buttons: New, Delete, Duplicate, Help, OK, Cancel

Microinstruction Halt(stop)

The screenshot shows the 'Edit machine instructions' dialog box. On the left, a table lists microinstructions with their names, opcodes, and field lengths. The 'stop' instruction is highlighted. In the center, the 'stop's implementation' is shown as a sequence of microinstructions: 'set-halt-bit' and 'End'. On the right, a list of 'Existing micros' is shown, including 'TransferRtoR', 'TransferAtoR', 'TransferRtoA', 'Set', 'Test', 'Increment', 'Arithmetic', 'Shift', 'Branch', 'Logical', 'Decode', 'MemoryAccess', 'IO', and 'SetCondBit'. Buttons for '<<insert<<', '>>delete', 'New', 'Delete', 'Duplicate', 'Help', 'OK', and 'Cancel' are visible.

name	opcode	fieldLengths
Add	C	4 12
stop	0	16
load	1	4 12
store	2	4 12
read	3	4 (12)
write	4	4 (12)
add	5	4 12
subtract	6	4 12
divide	8	4 12
jump	9	4 12
jmpz	A	4 12
jmpn	B	4 12

stop's implementation

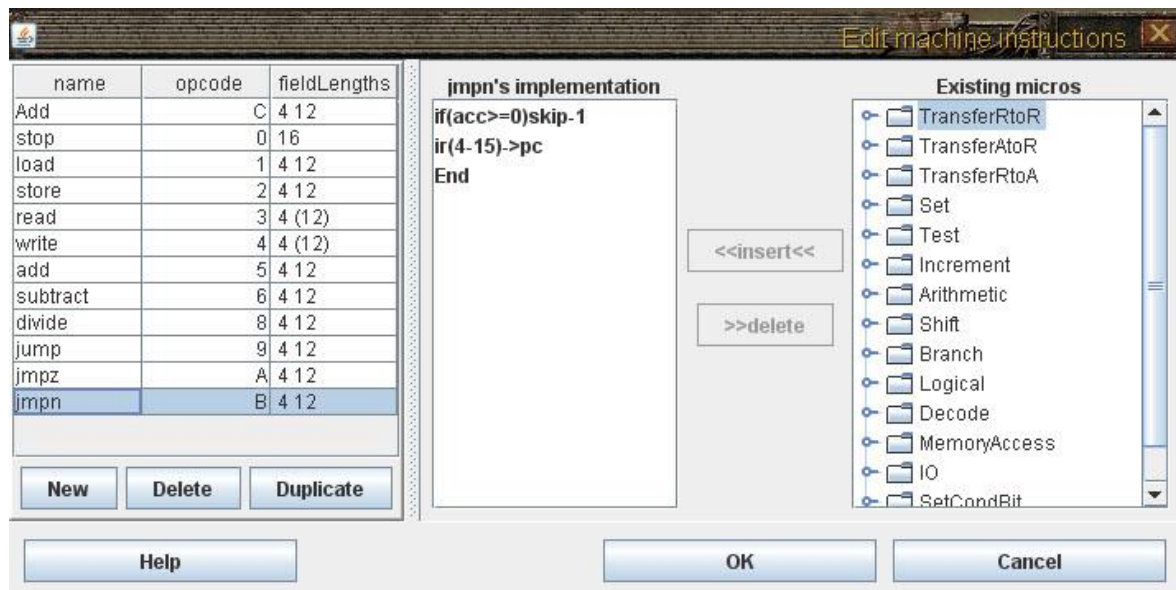
```
set-halt-bit
End
```

Existing micros

- TransferRtoR
- TransferAtoR
- TransferRtoA
- Set
- Test
- Increment
- Arithmetic
- Shift
- Branch
- Logical
- Decode
- MemoryAccess
- IO
- SetCondBit

Buttons: New, Delete, Duplicate, Help, OK, Cancel

Microinstruction Jmpn



ANSWER QUESTION (3):

